UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/708,722	11/09/2000	Stephan J. Jourdan	2207/9800	2194

25693   7590   02/18/2009
KENYON & KENYON LLP
RIVERPARK TOWERS, SUITE 600
333 W. SAN CARLOS ST.
SAN JOSE, CA 95110

EXAMINER
LI, AIMEE J

ART UNIT	PAPER NUMBER
2183	

MAIL DATE	DELIVERY MODE
02/18/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

	Application No.	Applicant(s)
Office Action Summary	09/708,722	JOURDAN ET AL.
	Examiner	Art Unit	
	AIMEE J. LI	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on <u>12 December 2008</u>.

2a) ☐ This action is **FINAL**.   2b) ☒ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) <u>1-19</u> is/are pending in the application.

     4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) <u>1-19</u> is/are rejected.

7) ☐ Claim(s) _____ is/are objected to.

8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☒ The drawing(s) filed on <u>24 February 2004</u> is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.

     Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

     Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

     a) ☐ All   b) ☐ Some *   c) ☐ None of:

         1. ☐ Certified copies of the priority documents have been received.

         2. ☐ Certified copies of the priority documents have been received in Application No. _____.

         3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

     * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail. Date. _____.

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1.   Claims 1-19 have been considered. Claims 1, 5, 8, and 14 have been amended as per

Applicants' request.

Papers Submitted

2.   It is hereby acknowledged that the following papers have been received and placed of

record in the file: RCE as filed 12 December 2008; Extension of Time for 3 Months as filed 12

December 2008; and Amendment as filed 12 December 2008.

Continued Examination Under 37 CFR 1.114

3.   A request for continued examination under 37 CFR 1.114, including the fee set forth in

37 CFR 1.17(e), was filed in this application after final rejection. Since this application is

eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e)

has been timely paid, the finality of the previous Office action has been withdrawn pursuant to

37 CFR 1.114. Applicant's submission filed on 12 December 2008 has been entered.

Claim Rejections - 35 USC § 102

4.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the

basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on
> sale in this country, more than one year prior to the date of application for patent in the United States.

5.   Claims 1-5, 8, and 11-14 are rejected under 35 U.S.C. 102(b) as being taught by Peleg et

al., U.S. Patent Number 5,381,533 (herein referred to as Peleg). Examiner notes that this patent

was cited by Applicants in the IDS filed 29 December 2000.

6.   Referring to claim 1, Peleg has taught a cache comprising:

a.   a cache line to store an instruction segment further comprising a plurality of

instructions stored in sequential positions of cache line in reverse program order

(Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8,

line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column

11, line 12; Figure 5A; Figure 5B; Figures 6-12 – In regards to Peleg, the cache

stores the basic blocks in the order they are executed, including, as shown in

Figure 10, when the basic blocks go backwards, i.e. reverse order.),

b.   wherein a conditional branch causing program flow to jump from a first location

in a first instruction stream to a second location in a second instruction stream

causes a terminal instruction from the second instruction stream to be stored in a

first position of a cache line (Peleg column 1, line 64 to column 2, line 19; column

4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24;

column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12 – In

regards to Peleg, a basic block starts with the instruction executed after a branch

and terminates with a branch instructions. This means that when a basic block's

first instruction is a branch instruction, it terminates with that instruction, and the

basic block only consists of the one branch instruction. So, when the single

instruction basic block is jumped to, that single beginning and ending instruction

is stored in the cache.).

7.   Referring to claim 2, Peleg has taught the cache of claim 1, wherein the instruction

segment is an extended block (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-

23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12).

8.   Referring to claim 3, Peleg has taught the cache of claim 1, wherein the instruction segment is a trace (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12).

9.   Referring to claim 4, Peleg has taught the cache of claim 1, wherein the instruction segment is a basic block (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12).

10.   Referring to claim 5, Peleg has taught a segment cache for a front-end system in a processor, comprising

        a.   a plurality of cache entries to store instructions of instruction segments in reverse program order (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12 – In regards to Peleg, the cache stores the basic blocks in the order they are executed, including, as shown in Figure 10, when the basic blocks go backwards, i.e. reverse order.),

        b.   wherein a conditional branch causing program flow to jump from a first location in a first instruction stream to a second location in a second instruction stream causes a terminal instruction from the second instruction stream to be stored in a first position of a cache line (Peleg column 1, line 64 to column 2, line 19; column

4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24;

column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12 – In

regards to Peleg, a basic block starts with the instruction executed after a branch

and terminates with a branch instructions. This means that when a basic block's

first instruction is a branch instruction, it terminates with that instruction, and the

basic block only consists of the one branch instruction. So, when the single

instruction basic block is jumped to, that single beginning and ending instruction

is stored in the cache.).

11.   Referring to claim 8, Peleg has taught a method comprising:

    a.   building an instruction segment based on program flow (Peleg column 1, line 64

        to column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line

        35; column 10, lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A;

        Figure 5B; Figures 6-12), and

    b.   storing instructions of the instruction segment in a cache entry in reverse program

        order (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23;

        column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58

        to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12 – In regards to Peleg,

        the cache stores the basic blocks in the order they are executed, including, as

        shown in Figure 10, when the basic blocks go backwards, i.e. reverse order.),

    c.   wherein a conditional branch causing program flow to jump from a first location

        in a first instruction stream to a second location in a second instruction stream

        causes a terminal instruction from the second instruction stream to be stored in a

first position of a cache line (Peleg column 1, line 64 to column 2, line 19; column

4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24;

column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12 – In

regards to Peleg, a basic block starts with the instruction executed after a branch

and terminates with a branch instructions. This means that when a basic block's

first instruction is a branch instruction, it terminates with that instruction, and the

basic block only consists of the one branch instruction. So, when the single

instruction basic block is jumped to, that single beginning and ending instruction

is stored in the cache.).

12.   Referring to claim 11, Peleg has taught the method of claim 8, wherein the instruction

segment is an extended block (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-

23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column

11, line 12; Figure 5A; Figure 5B; Figures 6-12).

13.   Referring to claim 12, Peleg has taught the method of claim 8, wherein the instruction

segment is a trace (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23; column

8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11, line 12;

Figure 5A; Figure 5B; Figures 6-12).

14.   Referring to claim 13, Peleg has taught t he method of claim 8, wherein the instruction

segment is a basic block (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23;

column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10, line 58 to column 11,

line 12; Figure 5A; Figure 5B; Figures 6-12).

15.   Referring to claim 14, Peleg has taught a processing engine, comprising:

a.   a front end stage to build and store instruction segments, instructions provided

       therein in reverse program order (Peleg column 1, line 64 to column 2, line 19;

       column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-

       24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12

       – In regards to Peleg, the cache stores the basic blocks in the order they are

       executed, including, as shown in Figure 10, when the basic blocks go backwards,

       i.e. reverse order.),

b.   wherein a conditional branch causing program flow to jump from a first location

       in a first instruction stream to a second location in a second instruction stream

       causes a terminal instruction from the second instruction stream to be stored in a

       first position of a cache line (Peleg column 1, line 64 to column 2, line 19; column

       4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24;

       column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12 – In

       regards to Peleg, a basic block starts with the instruction executed after a branch

       and terminates with a branch instructions. This means that when a basic block's

       first instruction is a branch instruction, it terminates with that instruction, and the

       basic block only consists of the one branch instruction. So, when the single

       instruction basic block is jumped to, that single beginning and ending instruction

       is stored in the cache.), and

c.   an execution unit in communication with the front end stage. (Peleg Figure 1).

Claim Rejections - 35 USC § 103

16.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in
> section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are
> such that the subject matter as a whole would have been obvious at the time the invention was made to a person
> having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the
> manner in which the invention was made.

17.   Claims 6-7 and 15-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Peleg et al., U.S. Patent Number 5,381,533 (herein referred to as Peleg) as applied to claims 5

and 14 above, and further in view of Rotenberg et al.'s "A Trace Cache Microarchitecture and

Evaluation" IEEE ©1999 (herein referred to as Rotenberg).

18.   Referring to claim 6, Peleg has taught the segment cache of claim 5, further comprising:

    a.   an instruction cache system (Peleg column 1, line 64 to column 2, line 19; column

          4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24;

          column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12),

    b.   an instruction segment system, comprising:

        i.   a fill unit provided in communication with the instruction cache system

              (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23;

              column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10,

              line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12).

19.   Peleg has not taught a selector coupled to an output of the instruction cache system and to

an output of the segment cache. Rotenberg has taught a selector coupled to an output of the

instruction cache system and to an output of the segment cache (Rotenberg Section 2.1 Trace-

Level Sequencing "…The output of the trace cache is one or more traces…"; Section 2.2

Instruction-Level Sequencing "The *outstanding trace buffers* in Fig. 2 are used to 1) construct

new traces that are not in the trace cache and 2) track branch outcomes…"; and Figure 2 – In

regards to Rotenberg, Figure 2 shows the output of the trace cache and the outstanding trace

buffers are connected to a line that has two inputs and one output, which is a selector. The

selector chooses between and existing trace in the trace cache, i.e. a trace cache hit, or a newly

formed trace, i.e. a trace cache miss or misprediction.) A person of ordinary skill in the art at the

time the invention was made, and as taught by Rotenberg, would have recognized that the trace

cache system provides fast trace-level sequencing while providing a method to create nonexistent

traces or repair mispredicted traces (Rotenberg Section 2.1 Trace-Level Sequencing "…The trace

predictor and trace cache together provide fast trace-level sequencing…Instruction-level

sequencing, discussed in the next section, is required to construct nonexistent traces or repair

trace mispredictions."). Therefore, it would have been obvious to a person of ordinary skill in

the art at the time the invention was made to incorporate the trace caches, buffers, and selector of

Rotenberg in the device of Peleg to increase the speed of trace-level sequencing while using

instruction-level sequencing to create new or correct mispredicted traces.

20.   Referring to claim 7, Peleg has not taught an apparatus of claim 6, wherein the instruction

segment system further comprises a segment predictor provided in communication with the

segment cache. Rotenberg has taught an apparatus of claim 6, wherein the instruction segment

system further comprises a segment predictor provided in communication with the segment

cache (Rotenberg Section 2.1 Trace-Level Sequencing "…A *next trace predictor*[14] treats

traces as basic units and explicitly predicts sequences of traces…" and Figures 2). A person of

ordinary skill in the art at the time the invention was made, and as taught by Rotenberg, would

have recognized that the trace predictor achieves high branch prediction throughput with a single

prediction per cycle (Rotenberg Section 2.1 Trace-Level Sequencing "…high branch prediction

throughput is implicitly achieved with only a single trace prediction per cycle…"). Therefore, it

would have been obvious to a person of ordinary skill in the art at the time the invention was

made to incorporate the trace predictor of Rotenberg in the device of Peleg to achieve high

branch prediction throughput in a single prediction cycle.

21.   Referring to claim 15, Peleg has taught the processing engine of claim 14, wherein the

front-end stage comprises:

     a.   an instruction cache system (Peleg column 1, line 64 to column 2, line 19; column

             4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24;

             column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12),

     b.   an instruction segment system, comprising:

          i.   a fill unit provided in communication with the instruction cache system

                 (Peleg column 1, line 64 to column 2, line 19; column 4, lines 11-23;

                 column 8, line 30 to column 9, line 35; column 10, lines 1-24; column 10,

                 line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12),

          ii.   a segment cache (Peleg column 1, line 64 to column 2, line 19; column 4,

                 lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24;

                 column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures

                 6-12).

22.   Peleg has not taught a selector coupled to an output of the instruction cache system and to

an output of the segment cache. Rotenberg has taught a selector coupled to an output of the

instruction cache system and to an output of the segment cache (Rotenberg Section 2.1 Trace-

Level Sequencing "…The output of the trace cache is one or more traces…"; Section 2.2

Instruction-Level Sequencing "The *outstanding trace buffers* in Fig. 2 are used to 1) construct

new traces that are not in the trace cache and 2) track branch outcomes…"; and Figure 2 – In

regards to Rotenberg, Figure 2 shows the output of the trace cache and the outstanding trace

buffers are connected to a line that has two inputs and one output, which is a selector. The

selector chooses between and existing trace in the trace cache, i.e. a trace cache hit, or a newly

formed trace, i.e. a trace cache miss or misprediction.) A person of ordinary skill in the art at the

time the invention was made, and as taught by Rotenberg, would have recognized that the trace

cache system provides fast trace-level sequencing while providing a method to create nonexistent

traces or repair mispredicted traces (Rotenberg Section 2.1 Trace-Level Sequencing "…The trace

predictor and trace cache together provide fast trace-level sequencing…Instruction-level

sequencing, discussed in the next section, is required to construct nonexistent traces or repair

trace mispredictions."). Therefore, it would have been obvious to a person of ordinary skill in

the art at the time the invention was made to incorporate the trace caches, buffers, and selector of

Rotenberg in the device of Peleg to increase the speed of trace-level sequencing while using

instruction-level sequencing to create new or correct mispredicted traces.

23.   Referring to claim 16, Peleg in view of Rotenberg has taught the processing engine of

claim 15, wherein the instruction segments are extended blocks (Peleg column 1, line 64 to

column 2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10,

lines 1-24; column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12).

24.   Referring to claim 17, Peleg in view of Rotenberg has taught the processing engine of

claim 15, wherein the instruction segments are traces (Peleg column 1, line 64 to column 2, line

19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column

10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12).

25.   Referring to claim 18, Peleg in view of Rotenberg has taught the processing engine of

claim 15, wherein the instruction segments are basic blocks (Peleg column 1, line 64 to column

2, line 19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24;

column 10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12).

26.   Referring to claim 19, Peleg in view of Rotenberg has taught the processing engine of

claim 15, wherein the instruction segment cache system further comprises a segment predictor

provided in communication with the segment cache.

27.   Claims 9-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Peleg et al.,

U.S. Patent Number 5,381,533 (herein referred to as Peleg) as applied to claim 8 above, and

further in view of Kyker et al., U.S. Patent Number 6,578,138 (herein referred to as Kyker).

28.   Referring to claim 9, Peleg has taught the method of claim 8, further comprising building

a second instruction segment based on program flow (Peleg column 1, line 64 to column 2, line

19; column 4, lines 11-23; column 8, line 30 to column 9, line 35; column 10, lines 1-24; column

10, line 58 to column 11, line 12; Figure 5A; Figure 5B; Figures 6-12). Peleg has not taught if

the first and second instruction segments overlap, extending the first instruction segment to

include non-overlapping instructions from the second instruction segment. Kyker has taught if

the first and second instruction segments, overlap, extending the first instruction segment to

include non-overlapping instructions from the second instruction segment (Kyker Abstract "…a

cache unit, which includes a data array that stores traces…In one exemplary method of unrolling

loops, the processor or trace cache unrolls loops…"; column 2, line 59 to column 3, line 33

"…when the trace cache determines that a loop is present, the trace cache continues to build the

trace by building additional iterations of the loop until the trace is a minimal length…In other

words, the trace cache builds the loop repeatedly until the trace is, for example, over two trace

lines long…"; Figure 1; and Figure 2 – In regards to Kyker, as shown in Figure 1, the trace is

built as normal, e.g. a first instruction segment is built normally, when there is no backward-

taken branch. When a backward branch is taken to form another, second instruction segment,

another iteration of the loop is added to the trace, thereby extending the first segment to include

the first segment.). A person of ordinary skill in the art at the time the invention was made

would have recognized that unrolling loops for a single continuous block of instructions

improves processor efficiency by reducing the number of times the processor accesses the cache

and maximizing usage of a cache line. Therefore, it would have been obvious to a person of

ordinary skill in the art at the time the invention was made to incorporate the loop unrolling of

Kyker in the device of Peleg to improve efficiency.

29.   Referring to claim 10, Peleg in view of Kyker has taught the method of claim 9, wherein

the extending comprises storing the non-overlapping instructions in the cache in reverse program

order in successive cache positions adjacent to the instructions from the first instruction segment

(Kyker Abstract "…a cache unit, which includes a data array that stores traces…In one

exemplary method of unrolling loops, the processor or trace cache unrolls loops…"; column 2,

line 59 to column 3, line 33 "…when the trace cache determines that a loop is present, the trace

cache continues to build the trace by building additional iterations of the loop until the trace is a

minimal length…In other words, the trace cache builds the loop repeatedly until the trace is, for

example, over two trace lines long…"; Figure 1; and Figure 2 – In regards to Kyker, as shown in

Figure 2 and explained the in the cited lines, the loop has three main instructions and, when the

backwards branch is encountered, the program jumps backwards, i.e. reverses directions, to the

beginning of the loop instead of moving forward in the program. As shown in Figure 2, when L_H

is encountered, previously stored instructions L_2 and L_3 are stored again in consecutive lines in

the cache, e.g. the trace cache reverses the program direction and stores previously stored

instructions again in the trace cache when a loop is encountered.).

Response to Arguments

30.   Applicant's arguments with respect to claims 1-19 have been considered but are moot in

view of the new ground(s) of rejection.

Conclusion

31.   The prior art made of record and not relied upon is considered pertinent to applicant's

disclosure.

  a. Rodgers et al., U.S. Patent Number 6,496,925, has taught a trace cache with

    branch prediction.

  b. Kalafatis et al., U.S. Patent Numbers 6,535,905 and 6,785,890, has taught a trace

    cache and a trace branch predictor.

  c. Davies et al., U.S. Patent Number 6,961,847, has taught a trace cache with its own

    branch predictor.

32.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to AIMEE J. LI whose telephone number is (571)272-4169. The

examiner can normally be reached on M-T 7:00am-4:30pm.

33.   If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the

organization where this application or proceeding is assigned is 571-273-8300.

34.   Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would

like assistance from a USPTO Customer Service Representative or access to the automated

information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Aimee J Li/
Primary Examiner, Art Unit 2183

15 February 2009